Project 3: Generating Anime Faces Using DCGANs and LLMs

CSCI 297: Generative AI

Due: March 27th, 2024, at 11:59PM

Overview

This assignment focuses on Deep Convolutional Generative Adversarial Networks (DCGANs) to generate images of anime faces items using PyTorch. The core objective is to demystify the complexities of DCGANs and provide hands-on experience in implementing and training these models to produce high-quality, diverse images of fashion items.

The assignment is comprised of two main components: the generator and the discriminator. You will design the generator to transform random noise into images resembling the Anime Faces dataset, while the discriminator will learn to distinguish between real and generated images. This adversarial process is at the heart of DCGAN training and is central to understanding how generative models learn.

Here is where things get interesting, you cannot write any code (muhahahahaha). You need to use Claude3, ChatGPT or some other LLM to generate the code for you. I want you to experience what dependency on these tools looks like. I want to reiterate that you are not allowed to write a single line of code for this assignment. If you are unable to get your code to work, you must figure out where your LLM went wrong and what it was unable to generate correctly. You will include a report on whether the LLM was able to generate something correctly or not. You will not lose points so long as you document where the model went wrong.

You will turn in a comprehensive Jupyter notebook and a report, will not only demonstrate the technical proficiency in implementing DCGANs but also encourage critical thinking about model architecture choices, training stability, and the potential applications and ethical considerations of generative models.

Implementation Steps

1. Dataset Preparation
    a. Ensure that you preprocess your dataset such that your DCGAN can accept the images as input.

2. DCGAN Architecture
    a. Design the generator network to transform random noise vectors into anime face images.
    b. Construct the discriminator network to differentiate between real images from the dataset and fake images produced by the generator.
    c. Ensure that both networks adhere to the DCGAN architecture principles.

3. Loss and Optimizer
    a. Define appropriate loss functions for both the generator and discriminator.
    b. Choose optimizers that facilitate stable and efficient training of DCGANs.

4. Training Loop
    a. Develop the training loop where the generator and discriminator are trained in an adversarial manner, ensuring the effective learning of both networks.
    b. Implement techniques to monitor and stabilize the training process.

5. Visualization
    a. Create visualizations to monitor the progress of generated images over training epochs.
    b. Visualize loss curves to analyze the training dynamics and stability.

6. Evaluation
    a. Evaluate the performance of the trained DCGAN model, focusing on the quality and diversity of generated images.

Turn In

- A Jupyter notebook containing the implemented PyTorch code for the GANs, along with inline comments explaining the steps.
- A report discussing the model architecture, training process, encountered challenges, and how they were addressed. Your report should also include the shortcomings of the LLM and what you were unable to get working.
- Visualizations of generated images and loss curves.